

Working with Script Files

This chapter covers the following topics:

- Executing a Script File
 - Defining User Variables
 - Aborting or Terminating a Script File
 - Debugging a Script File
-

Executing a Script File

Script files are executed from the viewers. Before a script file can be executed, it must be copied to the subfolder *Production*. See *Script File Folders* in the *Script Files* documentation.

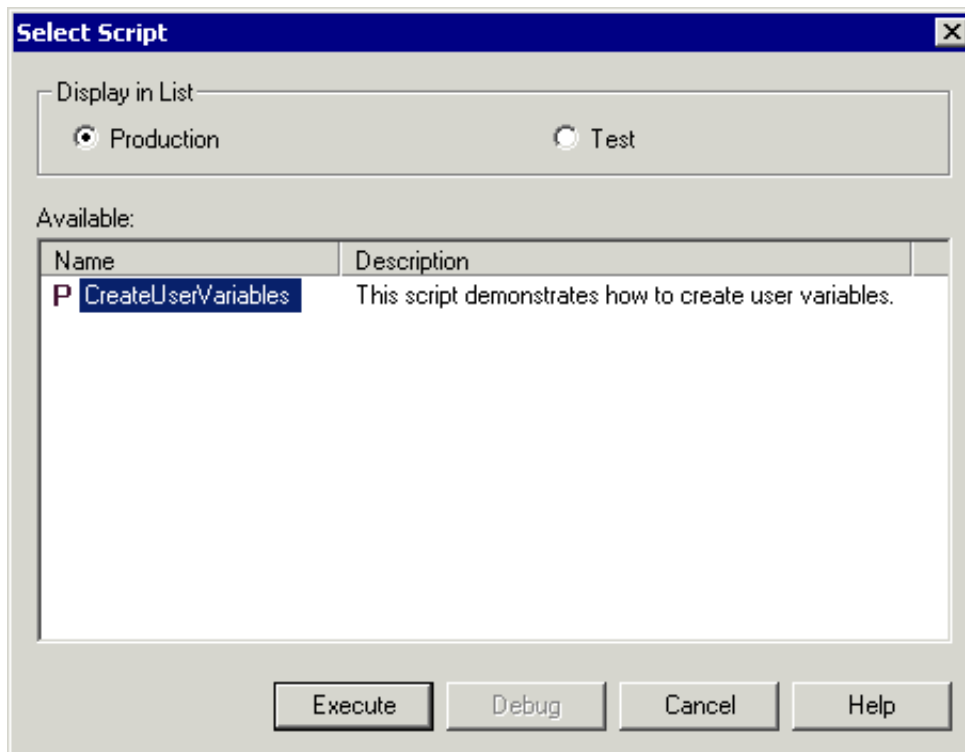
To execute a script file

1. From the **Utilities** menu, choose **Script List**.

Note:

When working in embedded mode with the Web Viewer, the above command is available from the context menu which appears when you click the right mouse button.

The Select Script dialog box appears only when script files have been copied to the script files folder. Otherwise a message appears indicating that no script files have been found.



The dialog box contains all script files that are stored in the subfolder *Production*.

Note:

The script files stored in the subfolder *Test* can only be shown when working with the Terminal Viewer. For all other types of viewer, the corresponding option button is always disabled. The **Debug** button is only available from the Terminal Viewer and only for script files in the subfolder *Test*. For all other types of viewer and for script files in the subfolder *Production*, this button is disabled. See *Debugging a Script File* for further information.

2. Select the desired script file.
3. Choose the **Execute** button.

Defining User Variables

When you run a script file which uses the user interface method `SetUserVariableValue`, you can set and change your own values for the user variables. If such a script file has not been run, the Define User Variables dialog box (see below) is empty and it is not possible to define user variables.

For example, you can specify your own password for mainframe access. All user variable values will be stored in encrypted format.

This feature is only available when a user is logged on. If anonymous logon has been defined, it is not possible to define user variables.

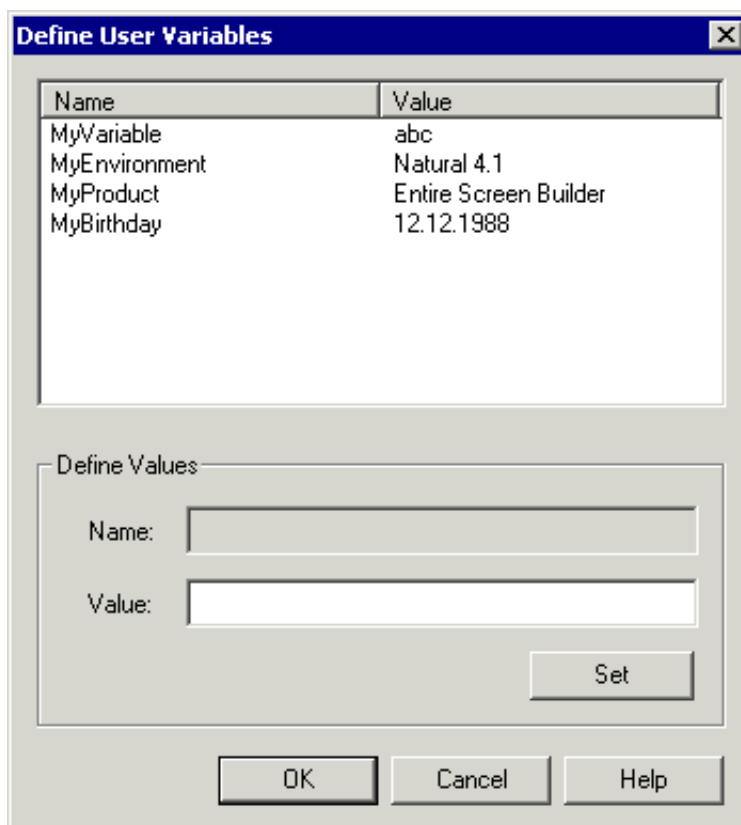
► To modify the value of a variable

1. From the **Utilities** menu, choose **User Variables**.

Note:

When working in embedded mode with the Web Viewer, the above command is available from the context menu which appears when you click the right mouse button.

The Define User Variables dialog box appears. The user variables are shown in the upper part of the dialog box.



2. Select a user variable.

3. Enter the new value for this user variable in the **Value** text box.
4. Choose the **Set** button.
5. Choose the **OK** button to close the dialog box.

Aborting or Terminating a Script File

The execution of a script files is aborted or terminated in the following cases:

- When you choose the **Cancel** button in a dialog box that was invoked by the script file.
- When a non-recoverable error occurs during script file processing (for example, a syntax or logic error).
- When the `Cancel` method is issued from within the script file.
- When a normal exit from the script file occurs (i.e. when the last statement is executed).

You can also abort the currently active script file as described below.

To abort the currently active script file

- From the **Utilities** menu, choose **Cancel Script**.

Note:

When working in embedded mode with the Web Viewer, the above command is available from the context menu which appears when you click the right mouse button.

Or:

Choose the following toolbar button (Terminal Viewer only):



Debugging a Script File

This feature is only available in the Terminal Viewer. It is only available when it has been allowed by the administrator (see *Users* in Entire Screen Builder's *System Management Hub* documentation).

Caution:

Debugging uses and blocks a lot of CPU power and resources on the Entire Screen Builder Server. We recommend not to debug script files in a production environment.

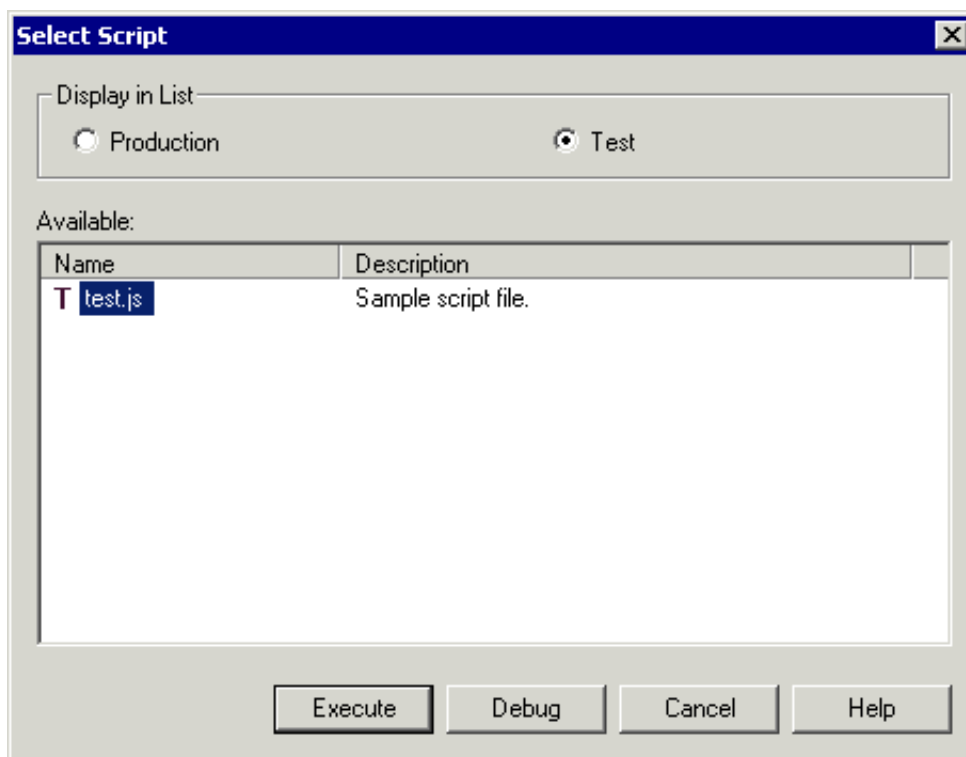
Before a script file can be debugged, it must be copied to the subfolder *Test*.

► To debug a script file

1. From the **Utilities** menu, choose **Script List** to invoke the Select Script dialog box.

See also: *Executing a Script File*.

2. Choose the **Test** option button to display the script files that can be debugged.



3. Select the script file to be debugged.
4. Choose the **Debug** button.
5. Use the **Debug** menu or debug toolbar to debug the script file (see below).
6. When debugging has been completed, terminate the debug session using the corresponding command.

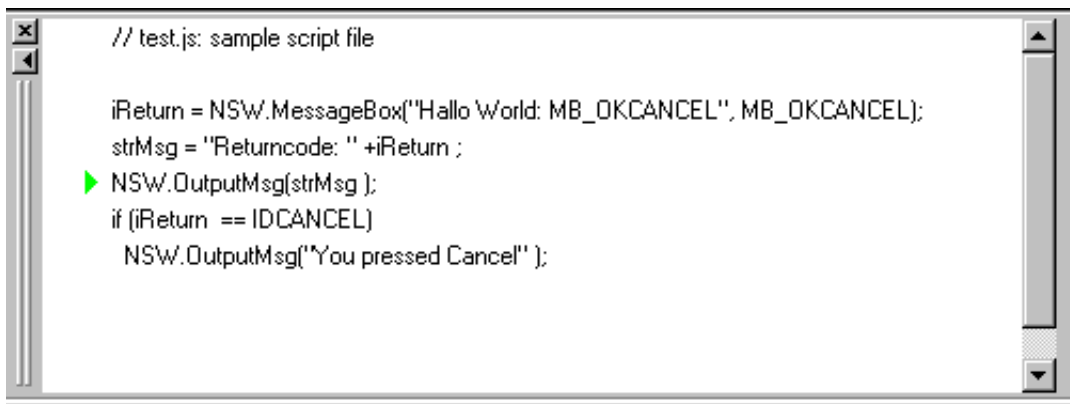
When debugging a script file, the following additional elements are shown in the Terminal Viewer:

- The **Debug** menu. This menu is only available as long as you debug a script file. It provides the same commands as the debug toolbar.
- A debug toolbar. This toolbar is only available as long as you debug a script file.

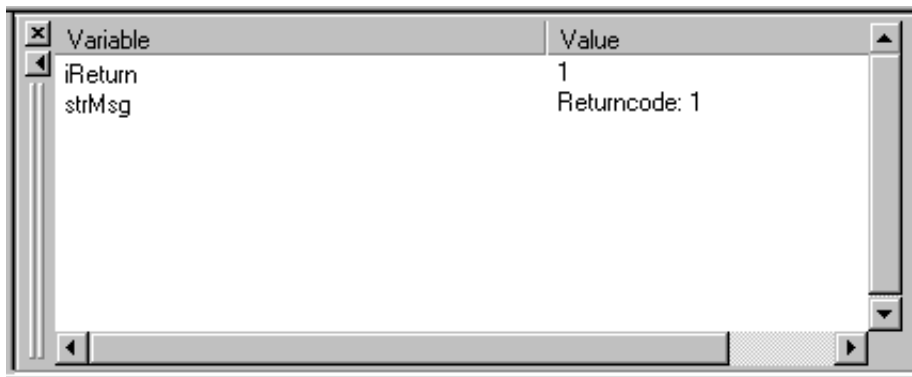
You can move it to the application window (e.g. below the menu bar or next to the terminal emulation screen) so that it is no longer shown in a window. To prevent docking, press CTRL while moving the window.



- A source window containing the source code on the left.










- A variables window on the right, showing the variables used in the script file.



You can modify the size of the source and variables windows by dragging a window border with the mouse. Using the mouse, you can drag these windows to other positions. When both windows are docked, you can click one of the buttons showing an arrow to expand or contract the corresponding window in the application window.

Using the following buttons in the debug toolbar or key combinations, you debug the script file.

	CTRL+SHIFT+R	Run script.
	CTRL+SHIFT+S	Step script.
	CTRL+SHIFT+A	Animate script.
	CTRL+SHIFT+P	Toggle breakpoint.
	CTRL+SHIFT+B	Break script.
	CTRL+SHIFT+E	Set echo time. A dialog box appears, in which you can specify the desired delay in milliseconds before execution of the next instruction.
	CTRL+SHIFT+T	Terminate debugging.